UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/800,493	03/15/2004	Stephen Fife Sheldon	11466	9307

26890 7590 02/02/2009
JAMES M. STOVER
TERADATA CORPORATION
2835 MIAMI VILLAGE DRIVE
MIAMISBURG, OH 45342

EXAMINER	
SANDERS, AARON J	

ART UNIT	PAPER NUMBER
2168	

MAIL DATE	DELIVERY MODE
02/02/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/800,493
Filing Date: March 15, 2004
Appellant(s): SHELDON ET AL.

HOWARD L. SPEIGHT
REGISTRATION No. 37,733
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 7 November 2008 appealing from the Office action mailed 28 April 2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The Examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The Appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

Appellant's brief presents arguments relating to the objections to the drawings. This issue relates to petitionable subject matter under 37 CFR 1.181 and not to appealable subject

matter. See MPEP § 1002 and § 1201. The Appellant's statement of the grounds of rejection to be reviewed on appeal is otherwise correct.

(7) Claims Appendix

A substantially correct copy of appealed claim 15 appears on page 12 of the Appendix to the Appellant's brief. The minor errors are as follows: the limitation "perform expression optimization on the expressions" should be "perform expression optimization on one or more of the expressions," according to the claims filed 5 February 2008 and finally rejected 28 April 2008.

(8) Evidence Relied Upon

6,665,664	PAULLEY et al.	12-2003
2005/0055338	WARNER et al.	3-2005

Nuutila, Esko, "Transitive Closure," Helsinki University of Technology, October 9, 1995 <http://www.cs.hut.fi/~enu/tc.html>.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 112, First Paragraph

The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it

pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 1, 15, and 29 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

Claims 1, 15, and 29 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Specifically, the limitations "performing further query optimization to produce a result" and "saving the result in a memory" are not enabled and do not appear in the specification.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1, 7-14, 29, and 35-42 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The process of claim 1 is not statutory because abstract ideas alone are not patentable. To be patentable, a process must have a practical application and (1) be tied to a particular machine or (2) transform a particular article into a different state. *In re Comiskey*, 499 F.3d 1365, 1376-77 (Fed. Cir. 2007); *In re Bilski*, __ F.3d __ (Fed. Cir. 2008).

An algorithm that is only useful in connection with a computer is still not "tied" to a machine. *Gottschalk v. Benson*, 409 U.S. 63, 64, 71-72 (A method of converting binary-coded decimal numerals into pure binary numerals was "not limited to any particular art or technology, to any particular apparatus or machinery, or to any particular end use" and would "wholly preempt the mathematical formula and in practical effect would be a patent on the algorithm itself"). Rather, a claim reciting an algorithm is statutory only if, as employed in the process, "it is embodied in, operates on, transforms, or otherwise involves another class of statutory subject matter, i.e., a machine, manufacture, or composition of matter." *In re Comiskey*, 499 F.3d at 1376.

Here, the process could be performed by a human writing on a piece of paper. Although it may only be useful in connection with a computer, it is not tied to a "particular" machine, and wholly preempts the algorithm, as in *Benson*. It also does not transform a "particular" article into another state, since numbers and expressions are not physical articles.

As per claims 29 and 35-42, the instant claims are not a proper system claim because the recited process is not "tied" to the CPUs or data storage facilities. The CPUs or data storage facilities are at best for use with the claimed method. As such, the instant claims are non-statutory.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person

having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 7-9, 11-13, 15, 21-23, 25-27, 29, 35-37, and 39-41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Paulley et al., U.S. 6,665,664 ("Paulley"), in view of Warner et al., U.S. 2005/0055338 ("Warner").

1. Paulley teaches "*A method of processing a database query, the query including an expression, the method including*," see col. 1, ll. 27-31, "information processing environments and, more particularly, to computer-implemented methodologies for query optimization."

Paulley teaches "*performing expression optimization the expression*," see col. 7, ll. 42-55, "a preprocessing phase, in which expressions are simplified whenever possible."

Paulley teaches "*performing further query optimization to produce a result*," see col. 7, ll. 42-55, "a normalization phase, in which the simplified expression is analyzed and either fully converted to conjunctive normal form."

Paulley teaches "*saving the result in a memory*," see col. 8, ll. 5-13, "the normalization method of the present invention saves only the useful prime implicates it can derive from the original input."

Paulley teaches "*performing expression optimization before further query optimization*," see col. 7, ll. 42-55, "This preprocessing phase includes several steps that are designed to simplify the original query expression, thereby simplifying the matrix processing occurring in the normalization phase."

Paulley teaches "*and where the expression includes a sub-expression ("SE")*," see col. 13, ll. 1-13, "The present invention repeatedly generates prime implicates of disjunctive sub-

expressions nested within a conjunctive expression, thereby normalizing the search condition piece-by-piece."

Paulley teaches "*and where the expression optimization includes: representing the query as a tree structure*," see col. 11, ll. 56-65, "the SQL statements are passed to the parser 361 which converts the statements into a query tree."

Paulley teaches "*representing the expression in the tree structure as a parent node having a first child node and a second child node*," see col. 11, ll. 56-65, "the SQL statements are passed to the parser 361 which converts the statements into a query tree – a binary tree data structure which represents the components of the query."

Paulley does not teach "*where the first child node represents the sub-expression*." Warner does, however, see Fig. 1 and par. 5, "Drilling down lower into this expression tree, it can be seen that branching into node 104 from the left side is the child node 108 representing the 'A' value," where it is obvious that the referenced node "A" could be a "sub-expression." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Warner's teachings would have allowed Paulley's method and system to gain a standard means for representing expressions, see par. 5.

Paulley does not teach "*where the second child node represents the portion of the expression that is not the sub-expression*." Warner does, however, see Fig. 1 and par. 5, "Branching into the node 104 from the right side is the child node 110 representing the 'B' value," where the referenced node "B" is "not the sub-expression." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the

teachings of the cited references because Warner's teachings would have allowed Paulley's method and system to gain a standard means for representing expressions, see par. 5.

Paulley does not teach "*and where the parent node represents an operation between the first child node and the second child node*." Warner does, however, see Fig. 1 and par. 5, "Extending into node 102 from the left side is the output from the child node 104 representing the '+' operator," where the claimed "parent node" is the referenced "node 104." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Warner's teachings would have allowed Paulley's method and system to gain a standard means for representing expressions, see par. 5.

Paulley does not teach "*determining that the second child node represents the constant 0 and that the parent node represents an arithmetic operation selected from the group consisting of addition and subtraction*." Warner does, however, see Fig. 1 and par. 5, "Extending into node 102 from the left side is the output from the child node 104 representing the '+' operator... Branching into the node 104 from the right side is the child node 110 representing the 'B' value," where it is obvious that the referenced "B" could have the value "0." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Warner's teachings would have allowed Paulley's method and system to gain a standard means for representing expressions, see par. 5.

Paulley does not teach "*and in response, removing the parent node and its children from the tree structure and inserting the first child node in its place*." Warner does, however, see Figs. 1, 3A-C, and par. 6, "The intermediate result from the operator at node 104 is propagated upwards to be evaluated by the operator at node 102 along with the input from 106." If the

subtree 1 + 2 in Fig. 3C were instead the subtree 1 + 0, element 314 would be the value "1" instead of "3" and rather than storing the value "1" in a temporary buffer, it would have been obvious to one of ordinary skill in the database art at the time of the invention to modify Warner's teachings because storing intermediate results into buffers (as Warner does with element 314) can be relatively memory expensive, see Warner par. 7.

7. Paulley teaches "*The method of claim l, where the query includes an assignment list clause and where one or more of the expressions are in the assignment list clause*," see col. 18, ll. 37-54, "A linked list of pointers is used to track which branches in the expression tree should be converted."

8. Paulley teaches "*The method of claim 1, where the query includes a WHERE clause, and where one or more of the expressions are in the WHERE clause*," see col. 2, Table 1, "SELECT name FROM employees WHERE sal=l 0,000."

9. Paulley teaches "*The method of claim 1, where further query optimization includes: determining a satisfiability of the database query*," see col. 2, ll. 55-63, "Conjunctive conditions are useful because they must each evaluate to true in order for the query's Where clause to be satisfied."

11. Paulley teaches "*The method of claim 1, where further query optimization includes: determining one or more plans for executing the query*," see col. 2, ll. 37-54, "a component called the optimizer determines the 'plan' or the best method of accessing the data to implement the SQL query."

12. Paulley teaches "*The method of claim 11, where one of the one or more plans includes: scanning a table to locate rows that satisfy one or more conditions; and summing one*

or more columns in the rows that satisfy the one or more conditions," see col. 4, ll. 11-43, "The usefulness of converting the search conditions to conjunctive normal form is that for a clause that consists of only a single predicate (i.e., not 'ORed with anything'), for any row in the result of that query that predicate must be true" where then summing the columns would have the same "true" result.

13. Paulley teaches "*The method of claim 1, where further query optimization includes: selecting an optimal plan from executing the database query*," see col. 12, ll. 7-16, "The optimizer, therefore, performs an analysis of the query and picks the best execution plan, which in turn results in particular ones of the access methods being invoked during query execution."

15. Paulley teaches "*A computer program, stored on a tangible storage medium, for use in processing a database query, the query including an expression, the computer program including executable instructions that cause a computer to*," see col. 1, ll. 27-31, "information processing environments and, more particularly, to computer-implemented methodologies for query optimization."

Paulley teaches "*perform expression optimization on one or more of the expressions*," see col. 7, ll. 42-55, "a preprocessing phase, in which expressions are simplified whenever possible."

Paulley teaches "*perform further query optimization to produce a result*," see col. 7, ll. 42-55, "a normalization phase, in which the simplified expression is analyzed and either fully converted to conjunctive normal form."

Paulley teaches "*save the result in a memory*," see col. 8, ll. 5-13, "the normalization method of the present invention saves only the useful prime implicates it can derive from the original input."

Paulley teaches "*where the expression includes a sub-expression ("SE")*," see col. 13, ll. 1-13, "The present invention repeatedly generates prime implicates of disjunctive sub-expressions nested within a conjunctive expression, thereby normalizing the search condition piece-by-piece."

Paulley teaches "*where expression optimization is performed before further query optimization*," see col. 7, ll. 42-55, "This preprocessing phase includes several steps that are designed to simplify the original query expression, thereby simplifying the matrix processing occurring in the normalization phase."

Paulley teaches "*and where the computer program includes executable instructions that cause a computer to: represent the query as a tree structure*," see col. 11, ll. 56-65, "the SQL statements are passed to the parser 361 which converts the statements into a query tree."

Paulley teaches "*represent the expression in the tree structure as a parent node having a first child node and a second child node*," see col. 11, ll. 56-65, "the SQL statements are passed to the parser 361 which converts the statements into a query tree – a binary tree data structure which represents the components of the query."

Paulley does not teach "*where the first child node represents the sub-expression*." Warner does, however, see Fig. 1 and par. 5, "Drilling down lower into this expression tree, it can be seen that branching into node 104 from the left side is the child node 108 representing the 'A' value," where it is obvious that the referenced node "A" could be a "sub-expression." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Warner's teachings would have allowed Paulley's method and system to gain a standard means for representing expressions, see par. 5.

Paulley does not teach "*where the second child node represents the portion of the expression that is not the sub-expression.*" Warner does, however, see Fig. 1 and par. 5, "Branching into the node 104 from the right side is the child node 110 representing the 'B' value," where the referenced node "B" is "not the sub-expression." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Warner's teachings would have allowed Paulley's method and system to gain a standard means for representing expressions, see par. 5.

Paulley does not teach "*and where the parent node represents an operation between the first child node and the second child node.*" Warner does, however, see Fig. 1 and par. 5, "Extending into node 102 from the left side is the output from the child node 104 representing the '+' operator," where the claimed "parent node" is the referenced "node 104." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Warner's teachings would have allowed Paulley's method and system to gain a standard means for representing expressions, see par. 5.

Paulley does not teach "*determine that the second child node represents the constant 0 and that the parent node represents an arithmetic operation selected from the group consisting of addition and subtraction.*" Warner does, however, see Fig. 1 and par. 5, "Extending into node 102 from the left side is the output from the child node 104 representing the '+' operator... Branching into the node 104 from the right side is the child node 110 representing the 'B' value," where it is obvious that the referenced "B" could have the value "0." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the

teachings of the cited references because Warner's teachings would have allowed Paulley's method and system to gain a standard means for representing expressions, see par. 5.

Paulley does not teach "*and in response, remove the parent node and its children from the tree structure and insert the first child node in its place.*" Warner does, however, see Figs. 1, 3A-C, and par. 6, "The intermediate result from the operator at node 104 is propagated upwards to be evaluated by the operator at node 102 along with the input from 106." If the subtree 1 + 2 in Fig. 3C were instead the subtree 1 + 0, element 314 would be the value "1" instead of "3" and rather than storing the value "1" in a temporary buffer, it would have been obvious to one of ordinary skill in the database art at the time of the invention to modify Warner's teachings because storing intermediate results into buffers (as Warner does with element 314) can be relatively memory expensive, see Warner par. 7.

21. Paulley teaches "*The computer program of claim 15, where the query includes an assignment list clause and where one or more of the expressions are in the assignment list clause*," see col. 18, ll. 37-54, "A linked list of pointers is used to track which branches in the expression tree should be converted."

22. Paulley teaches "*The computer program of claim 15, where the query includes a WHERE clause, and where one or more of the expressions are in the WHERE clause*," see col. 2, Table 1, "SELECT name FROM employees WHERE sal=1 0,000."

23. Paulley teaches "*The computer program of claim 15, where further query optimization includes: determining a satisfiability of the database query*," see col. 2, ll. 55-63, "Conjunctive conditions are useful because they must each evaluate to true in order for the query's Where clause to be satisfied."

25. Paulley teaches "*The computer program of claim 15, where further query optimization includes: determining one or more plans for executing the query,*" col. 2, ll. 37-54, "a component called the optimizer determines the 'plan' or the best method of accessing the data to implement the SQL query."

26. Paulley teaches "*The computer program of claim 25, where one of the one or more plans includes: scanning a table to locate rows that satisfy one or more conditions; and summing one or more columns in the rows that satisfy the one or more conditions,*" see col. 4, ll. 11-43, "The usefulness of converting the search conditions to conjunctive normal form is that for a clause that consists of only a single predicate (i.e., not 'ORed with anything'), for any row in the result of that query that predicate must be true" where then summing the columns would have the same "true" result.

27. Paulley teaches "*The computer program of claim 15, where further query optimization includes: selecting an optimal plan from executing the database query,*" see col. 12, ll. 7-16, "The optimizer, therefore, performs an analysis of the query and picks the best execution plan, which in turn results in particular ones of the access methods being invoked during query execution."

29. Paulley teaches "*A database system including: a massively parallel processing system including,*" see Fig. 3.

Paulley teaches "*one or more nodes,*" see Fig. 3.

Paulley teaches "*a plurality of CPUs, each of the one or more nodes providing access to one or more CPUs,*" see Fig. 3.

Paulley teaches "*a plurality of data storage facilities each of the one or more CPUs providing access to one or more data storage facilities*," see Fig. 3.

Paulley teaches "*a process for execution on the massively parallel processing system for processing a database query, the query including an expression, the process including*," see col. 1, ll. 27-31, "information processing environments and, more particularly, to computer-implemented methodologies for query optimization."

Paulley teaches "*performing expression optimization on the expression*," see col. 7, ll. 42-55, "a preprocessing phase, in which expressions are simplified whenever possible."

Paulley teaches "*performing further query optimization to produce a result*," see col. 7, ll. 42-55, "a normalization phase, in which the simplified expression is analyzed and either fully converted to conjunctive normal form."

Paulley teaches "*saving the result in a memory*," see col. 8, ll. 5-13, "the normalization method of the present invention saves only the useful prime implicates it can derive from the original input."

Paulley teaches "*where the expression optimization is performed before the further query optimization*," see col. 7, ll. 42-55, "This preprocessing phase includes several steps that are designed to simplify the original query expression, thereby simplifying the matrix processing occurring in the normalization phase."

Paulley teaches "*and where the expression includes a sub-expressions ("SE")*," see col. 13, ll. 1-13, "The present invention repeatedly generates prime implicates of disjunctive sub-expressions nested within a conjunctive expression, thereby normalizing the search condition piece-by-piece."

Paulley teaches "*and where expression optimization includes: representing the query as a tree structure*," see col. 11, ll. 56-65, "the SQL statements are passed to the parser 361 which converts the statements into a query tree."

Paulley teaches "*representing the expression in the tree structure as a parent node having a first child node and a second child node*," see col. 11, ll. 56-65, "the SQL statements are passed to the parser 361 which converts the statements into a query tree – a binary tree data structure which represents the components of the query."

Paulley does not teach "*where the first child node represents the sub-expression*." Warner does, however, see Fig. 1 and par. 5, "Drilling down lower into this expression tree, it can be seen that branching into node 104 from the left side is the child node 108 representing the 'A' value," where it is obvious that the referenced node "A" could be a "sub-expression." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Warner's teachings would have allowed Paulley's method and system to gain a standard means for representing expressions, see par. 5.

Paulley does not teach "*where the second child node represents the portion of the expression that is not the sub-expression*." Warner does, however, see Fig. 1 and par. 5, "Branching into the node 104 from the right side is the child node 110 representing the 'B' value," where the referenced node "B" is "not the sub-expression." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Warner's teachings would have allowed Paulley's method and system to gain a standard means for representing expressions, see par. 5.

Paulley does not teach "*and where the parent node represents an operation between the first child node and the second child node.*" Warner does, however, see Fig. 1 and par. 5, "Extending into node 102 from the left side is the output from the child node 104 representing the '+' operator," where the claimed "parent node" is the referenced "node 104." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Warner's teachings would have allowed Paulley's method and system to gain a standard means for representing expressions, see par. 5.

Paulley does not teach "*determining that the second child node represents the constant 0 and that the parent node represents an arithmetic operation selected from the group consisting of addition and subtraction.*" Warner does, however, see Fig. 1 and par. 5, "Extending into node 102 from the left side is the output from the child node 104 representing the '+' operator... Branching into the node 104 from the right side is the child node 110 representing the 'B' value," where it is obvious that the referenced "B" could have the value "0." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Warner's teachings would have allowed Paulley's method and system to gain a standard means for representing expressions, see par. 5.

Paulley does not teach "*and in response, removing the parent node and its children from the tree structure and inserting the first child node in its place.*" Warner does, however, see Figs. 1, 3A-C, and par. 6, "The intermediate result from the operator at node 104 is propagated upwards to be evaluated by the operator at node 102 along with the input from 106." If the subtree 1 + 2 in Fig. 3C were instead the subtree 1 + 0, element 314 would be the value "1" instead of "3" and rather than storing the value "1" in a temporary buffer, it would have been

obvious to one of ordinary skill in the database art at the time of the invention to modify Warner's teachings because storing intermediate results into buffers (as Warner does with element 314) can be relatively memory expensive, see Warner par. 7.

35. Paulley teaches "*The database system of claim 29, where the query includes an assignment list clause and where one or more of the expressions are in the assignment list clause*," see col. 18, ll. 37-54, "A linked list of pointers is used to track which branches in the expression tree should be converted."

36. Paulley teaches "*The database system of claim 29, where the query includes a WHERE clause, and where one or more of the expressions are in the WHERE clause*," see col. 2, Table 1, "SELECT name FROM employees WHERE sal=1 0,000."

37. Paulley teaches "*The database system of claim 29, where further query optimization includes: determining a satisfiability of the database query*," see col. 2, ll. 55-63, "Conjunctive conditions are useful because they must each evaluate to true in order for the query's Where clause to be satisfied."

39. Paulley teaches "*The database system of claim 29, where further query optimization includes: determining one or more plans for executing the query*," see col. 2, ll. 37-54, "a component called the optimizer determines the 'plan' or the best method of accessing the data to implement the SQL query."

40. Paulley teaches "*The database system of claim 39, where one of the one or more plans includes: scanning a table to locate rows that satisfy one or more conditions; and summing one or more columns in the rows that satisfy the one or more conditions*," see col. 4, ll. 11-43, "The usefulness of converting the search conditions to conjunctive normal form is that for a

clause that consists of only a single predicate (i.e., not 'ORed with anything'), for any row in the result of that query that predicate must be true" where then summing the columns would have the same "true" result.

41. Paulley teaches "*The database system of claim 29, where further query optimization includes: selecting an optimal plan from executing the database query,*" see col. 12, ll. 7-16, "The optimizer, therefore, performs an analysis of the query and picks the best execution plan, which in turn results in particular ones of the access methods being invoked during query execution").

Claims 10, 14, 24, 28, 38, and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Paulley et al., U.S. 6,665,664 ("Paulley"), in view of Warner et al., U.S. 2005/0055338 ("Warner"), and in view of Esko Nuutila, "Transitive Closure," Helsinki University of Technology, 9 October 1995 ("Nuutila").

10. Paulley does not teach "*The method of claim 1, where further query optimization includes: determining a transitive closure of the database query.*" Nuutila does, however, see "The transitive closure of G is a graph G+ = (V,E+) such that for all v,w in V there is an edge (v,w) in E+ if and only if there is a non-null path from v to w in G." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Nuutila's teachings would have allowed Paulley's method and system to gain lower cost in executing queries on a database, see Nuutila, "It is required, for instance, in the reachability analysis of transition networks representing distributed and parallel systems and in the construction of parsing automata in compiler construction."

14. Paulley teaches "*The method of claim l, where further query optimization includes two or more optimizations selected from the group consisting of: determining a satisfiability of the database query*," see col. 2, ll. 55-63, "Conjunctive conditions are useful because they must each evaluate to true in order for the query's Where clause to be satisfied."

Paulley teaches "*determining one or more plans for executing the query*," see col. 2, ll. 37-54, "a component called the optimizer determines the 'plan' or the best method of accessing the data to implement the SQL query."

Paulley teaches "*and selecting an optimal plan from executing the database query*," see col. 12, ll. 7-16, "The optimizer, therefore, performs an analysis of the query and picks the best execution plan, which in turn results in particular ones of the access methods being invoked during query execution."

Paulley does not teach "*determining a transitive closure of the database query*." Nuutila does, however, see "The transitive closure of G is a graph G+ = (V,E+) such that for all v,w in V there is an edge (v,w) in E+ if and only if there is a non-null path from v to w in G." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Nuutila's teachings would have allowed Paulley's method and system to gain lower cost in executing queries on a database, see Nuutila, "It is required, for instance, in the reachability analysis of transition networks representing distributed and parallel systems and in the construction of parsing automata in compiler construction."

24. Paulley does not teach "*The computer program of claim 15, where further query optimization includes: determining a transitive closure of the database query*." Nuutila does,

however, see "The transitive closure of G is a graph G+ = (V,E+) such that for all v,w in V there is an edge (v,w) in E+ if and only if there is a non-null path from v to w in G." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Nuutila's teachings would have allowed Paulley's method and system to gain lower cost in executing queries on a database, see Nuutila, "It is required, for instance, in the reachability analysis of transition networks representing distributed and parallel systems and in the construction of parsing automata in compiler construction."

28. Paulley teaches "*The computer program of claim 15, where further query optimization includes two or more optimizations selected from the group consisting of: determining a satisfiability of the database query*," see col. 2, ll. 55-63, "Conjunctive conditions are useful because they must each evaluate to true in order for the query's Where clause to be satisfied."

Paulley teaches "*determining one or more plans for executing the query*," see col. 2, ll. 37-54, "a component called the optimizer determines the 'plan' or the best method of accessing the data to implement the SQL query."

Paulley teaches "*and selecting an optimal plan from executing the database query*," see col. 12, ll. 7-16, "The optimizer, therefore, performs an analysis of the query and picks the best execution plan, which in turn results in particular ones of the access methods being invoked during query execution."

Paulley does not teach "*determining a transitive closure of the database query*." Nuutila does, however, see "The transitive closure of G is a graph G+ = (V,E+) such that for all v,w in V

there is an edge (v,w) in E+ if and only if there is a non-null path from v to w in G." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Nuutila's teachings would have allowed Paulley's method and system to gain lower cost in executing queries on a database, see Nuutila, "It is required, for instance, in the reachability analysis of transition networks representing distributed and parallel systems and in the construction of parsing automata in compiler construction."

38. Paulley does not teach "*The database system of claim 29, where further query optimization includes: determining a transitive closure of the database query.*" Nuutila does, however, see "The transitive closure of G is a graph G+ = (V,E+) such that for all v,w in V there is an edge (v,w) in E+ if and only if there is a non-null path from v to w in G." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Nuutila's teachings would have allowed Paulley's method and system to gain lower cost in executing queries on a database, see Nuutila, "It is required, for instance, in the reachability analysis of transition networks representing distributed and parallel systems and in the construction of parsing automata in compiler construction."

42. Paulley teaches "*The database system of claim 29, where further query optimization includes two or more optimizations selected from the group consisting of: determining a satisfiability of the database query,*" see col. 2, ll. 55-63, "Conjunctive conditions are useful because they must each evaluate to true in order for the query's Where clause to be satisfied."

Paulley teaches "*determining one or more plans for executing the query*," see col. 2, ll. 37-54, "a component called the optimizer determines the 'plan' or the best method of accessing the data to implement the SQL query."

Paulley teaches "*and selecting an optimal plan from executing the database query*," see col. 12, ll. 7-16, "The optimizer, therefore, performs an analysis of the query and picks the best execution plan, which in turn results in particular ones of the access methods being invoked during query execution."

Paulley does not teach "*determining a transitive closure of the database query*." Nuutila does, however, see "The transitive closure of G is a graph G+ = (V,E+) such that for all v,w in V there is an edge (v,w) in E+ if and only if there is a non-null path from v to w in G." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Nuutila's teachings would have allowed Paulley's method and system to gain lower cost in executing queries on a database, see Nuutila, "It is required, for instance, in the reachability analysis of transition networks representing distributed and parallel systems and in the construction of parsing automata in compiler construction."

(10) Response to Argument

There are minor objections to the claims and drawings that have not been corrected because the amendment after final filed 28 July 2008 was not entered.

I. The 35 U.S.C. 112 first paragraph rejections of claims 1, 15, and 29.

As per Applicant's argument that the limitations "performing further query optimization to produce a result" complies with 35 U.S.C. 112, the Examiner respectfully disagrees. The Examiner agrees that Fig. 3 and par. 20 describe "performing further query optimization to produce a result," where the "further query optimization" is performed by the "optimizer 325" and the "result" is the "executable steps." It appears that given this interpretation of the claims, the query optimization performed before "further" query optimization is the claimed "expression optimization," since the query includes an expression.

While the limitation may comply with the written description requirement, it is not enabled because producing a generic "result" is far broader in scope than producing "executable steps." One of ordinary skill in the art reading the claims would not know that "performing further query optimization to produce a result" actually means performing query optimization to produce executable steps. More likely, an artisan would assume that the "result" is the set of answers to the optimized query.

As per Applicant's argument that the limitations "saving the result in a memory" complies with 35 U.S.C. 112, the Examiner respectfully disagrees. While one of ordinary skill in the art would understand what saving executable steps in a memory means, the limitation is not enabled because "result" is not enabled. Further, the limitation does not comply with the written description requirement because it does not appear in the specification.

II. The 35 U.S.C. 101 rejections of claims 1, 7-14, 29, and 35-42.

The 35 U.S.C. 101 rejections of claims 15 and 21-28 have been withdrawn in light of *Bilski*.

As per Applicant's argument that the 35 U.S.C. 101 rejections should be withdrawn because the limitation "produce a result" is enabled and is a tangible result, the Examiner respectfully disagrees. First, the limitation "produce a result" is not enabled, as explained above. Second, the Applicant correctly notes that the "useful, concrete, and tangible" test is no longer applicable. Instead, the claims must comply with Bilski's machine-or-transformation test. They do not. The process of claim 1 could be performed by a human writing on a piece of paper. Although it may only be useful in connection with a computer, it is not tied to a "particular" machine, and wholly preempts the algorithm, as in *Benson*. It also does not transform a "particular" article into another state, since numbers and expressions are not physical articles or representations of physical articles.

As per the system of claims 29 and 35-42, claim 29 is not a proper system claim because the recited process is not tied to the CPUs or data storage facilities. At best, the CPUs or data storage facilities are for use with the claimed method. As such, the instant claims are non-statutory.

III. The 35 U.S.C. 103 rejections of claims 1, 7-9, 11-13, 15, 21-23, 25-27, 29, 35-37, and 39-41.

As per Applicant's argument that the limitation "removing and inserting" is not obvious in view of Warner's "propagating," the Examiner respectfully disagrees. The relevant limitation

is "*determining that the second child node represents the constant 0 and that the parent node represents an arithmetic operation selected from the group consisting of addition and subtraction; and in response, removing the parent node and its children from the tree structure and inserting the first child node in its place*." The Examiner cited Fig. 1 and pars. 5-6, "Extending into node 102 from the left side is the output from the child node 104 representing the '+' operator… Branching into the node 104 from the right side is the child node 110 representing the 'B' value… The intermediate result from the operator at node 104 is propagated upwards to be evaluated by the operator at node 102 along with the input from 106." The claimed "first child node" is the referenced node 108 ("A"), the claimed "second child node" is the referenced node 110 ("B") and the claimed "parent node" is the referenced node 104 ("+"). It is obvious that the referenced value "B" could be the claimed constant "0." The process of propagating the intermediate result is shown in Warner Figs. 3A-C.

Fig. 3A shows an example expression tree before processing. Fig. 3B shows the steps performed to process that expression. The step of processing node 308 shows that the subtree 1 + 2 is evaluated and the resulting value 3 is propagated to node 302. Fig. 3C shows that the value 3 (element 314) replaces the subtree 1 + 2 and becomes a child node of node 302. Element 314 is stored in a temporary buffer, see par. 50.

If the subtree 1 + 2 were instead the subtree 1 + 0 (as in Applicant's independent claims), element 314 would be the value "1" instead of "3." In that situation, rather than storing the value "1" in a temporary buffer, it would have been obvious to one of ordinary skill in the art at the time of the invention to "[remove] the parent node and its children from the tree structure and [insert] the first child node in its place" (as in Applicant's independent claims) because storing

intermediate results into buffers (as Warner does with element 314) can be relatively memory expensive, see Warner par. 7.

IV. The 35 U.S.C. 103 rejections of claims 10, 14, 24, 28, 38, and 42.

Applicant's assumption that claims 10, 14, 24, 28, 38, and 42 are rejected over Paulley et al., U.S. 6,665,664 ("Paulley"), in view of Warner et al., U.S. 2005/0055338 ("Warner"), and in view of Esko Nuutila, "Transitive Closure," Helsinki University of Technology, 9 October 1995 ("Nuutila") is correct. The rejections of claims 10, 14, 24, 28, 38, and 42 should be upheld in view of the rejections of independent claims 1, 15, and 29.

V. Objections to the drawings.

Objections to the drawings are not appealable. Rather, they should be petitioned. See 37 C.F.R. 1.113(a), "Petition may be taken to the Director in the case of objections or requirements not involved in the rejection of any claim."

The Examiner agrees that the steps of "performing expression optimization" and "performing further query optimization" appear in Fig. 3, steps 320 and 325. The step of "saving the result," however, does not. Producing "executable steps" is not equivalent to "saving the result." Further, the steps of "representing the query" and "representing the expression" do not appear in the drawings. Fig. 4, cited by Applicant, shows an expression tree, not the <u>steps</u> of "representing the query" and "representing the expression."

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the Examiner in the Related Appeals and Interferences section of this Examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Aaron Sanders/

Examiner, Art Unit 2168

21 January 2009

Conferees:

/Tim T. Vo/

Supervisory Patent Examiner, Art Unit 2168

/Eddie C. Lee/

Supervisory Patent Examiner, TC 2100